Daylight Savings Time

LOOPTIME addition

Thu, Apr 25, 2002

The local application `LOOPTIME` includes partial support for the daylight savings time adjustments that occur twice each year in most parts of the USA. Two parameters may be set ahead of time to signal the application of a date on which the adjustment should be made, one for the spring and the other for the fall. This note describes a method that was implemented to make this adjustment more automatic.

The basic approach is to have the application decide for itself when the correct day is for making the adjustment. The rule is that the date in the spring when we enter DST is the first Sunday in April. The date in the fall when we leave DST is the last Sunday in October. Changes were made to the code to automatically determine the appropriate calendar date for the change ahead of time and install the value into the proper parameter word, so that the support already included will take care of the rest of the job.

The first part for the solution was to find an algorithm for determining the day-of-week for any calendar date. Searching on the web yielded the following algorithm:

Given a numeric value for `month`, `day`, and (4-digit) `year`, the calculation to yield day-of week in `d`, where `d`=0 means Sunday, `d`=1 means Monday, etc.

```
a = (14 – month)/12
y = year – a
m = month + 12*a – 2
d = (day + y + y/4 – y/100 + y/400 + 31*m/12) mod 7
```

All divisions are to be done with integer arithmetic. As an example, consider today’s date, where `month` = 4, `day` = 25, and `year` = 2002.

```
a = 10/12 = 0
y = 2002 – 0 = 2002
m = 4 + 12*0 – 2 = 2
d = (25 + 2002 + 2002/4 – 2002/100 + 2002/400 + 31*2/12) mod 7
  = (25 + 2002 + 500 – 20 + 5 + 5) mod 7
  = 2517 mod 7 = 4, which correctly indicates Thursday.
```

When the program notices that a new day has occurred, it checks the current month. If it is March or October, and if it is Sunday, it considers the matter further. If it is March, it asks what day-of-week is April 1; if it is October, it asks what day-of-week is October 25. Based upon the result it gets, it can determine which day is the first Sunday in April, or which day is the last Sunday in October. If the associated parameter word is zero, it is changed to reflect that correct day. For this year, the first Sunday in April was April 7, so the spring parameter would have been set to `0x0407`. The last Sunday in October will be October 27, so the fall parameter value will be `0x1027`. (These representations are, of course, in BCD.) Then, using the logic that was already in place, at 2 am on that day, the application will make the appropriate modification (via increment or decrement) to the GMT Offset parameter. In the Central Time Zone, this value is –5 in the summer and –6 in the winter.